electronics

Article

A Novel Feature-Selection Method for Human Activity Recognition in Videos

Nadia Tweit [1,*], Muath A. Obaidat [2,*], Majdi Rawashdeh [3], Abdalraoof K. Bsoul [4] and Mohammed GH. Al Zamil [4,*]

[1] Department of Information Systems, Yarmouk University, Irbid 21163, Jordan
[2] Department of Computer Science, City University of New York, New York, NY 10019, USA
[3] Department of Business Information Technology, PSUT, Amman 11941, Jordan; m.rawashdeh@psut.edu.jo
[4] Department of Computer Science, Yarmouk University, Irbid 21163, Jordan; raoofbsoul@yu.edu.jo
* Correspondence: nadiafaisaltweit@yu.edu.jo (N.T.); muobaidat@ccny.cuny.edu (M.A.O.); mohammedz@yu.edu.jo (M.G.A.Z.)

Abstract: Human Activity Recognition (HAR) is the process of identifying human actions in a specific environment. Recognizing human activities from video streams is a challenging task due to problems such as background noise, partial occlusion, changes in scale, orientation, lighting, and the unstable capturing process. Such multi-dimensional and none-linear process increases the complexity, making traditional solutions inefficient in terms of several performance indicators such as accuracy, time, and memory. This paper proposes a technique to select a set of representative features that can accurately recognize human activities from video streams, while minimizing the recognition time and memory. The extracted features are projected on a canvas, which keeps the synchronization property of the spatiotemporal information. The proposed technique is developed to select the features that refer only to progression of changes. The original RGB frames are preprocessed using background subtraction to extract the subject. Then the activity pattern is extracted through the proposed Growth method. Three experiments were conducted; the first experiment was a baseline to compare the classification task using the original RGB features. The second experiment relied on classifying activities using the proposed feature-selection method. Finally, the third experiment provided a sensitivity analysis that compares between the effect of both techniques on time and memory resources. The results indicated that the proposed method outperformed original RBG feature-selection method in terms of accuracy, time, and memory requirements.

Keywords: human activity recognition; image processing; video processing; feature selection; deep learning; neural networks; data analysis

Citation: Tweit, N.; Obaidat, M.A.; Rawashdeh, M.; Bsoul, A.K.; Al Zamil, M.G. A Novel Feature-Selection Method for Human Activity Recognition in Videos. *Electronics* **2022**, *11*, 732. https://doi.org/10.3390/electronics11050732

Academic Editor: Byung Cheol Song

Received: 6 January 2022
Accepted: 23 February 2022
Published: 26 February 2022

1. Introduction

Video activity recognition is the process of identifying certain actions that represent an activity based on a collection of video-streams observations. Since the 1980s, this area has attracted the research community due to its realistic implementations in several fields such as healthcare surveillance systems [1,2], smart environments [3,4], surveillance and security systems for indoor and outdoor activities [5,6], and entertainment [7]. Video-based activity recognition is a challenging task due to the spatiotemporal aspect of frames. Specifically, consecutive video frames are dependent and reflect redundant information.

Although a video clip comprises a collection of 2D frames, action recognition from videos tends to be a complex image-based classification approach. Besides the spatial features and details found in a single frame, videos add an extra dimension of temporal aspects with essential motion features [8]. Some activities may simply be recognized by using a single image. Since single frame analysis can sometimes be ambiguous to distinguish an activity from others. Therefore, motion patterns are very essential to distinguish different activities.

The vast number of features that can be extracted from a video to recognize a human activity complicate the process, making traditional solutions inefficient in terms of several performance indicators such as accuracy, time, and memory. Recently, many attempts have been made to overcome this problem, which adopt several strategies such as Hidden Markov Models [9,10], optical flow-based techniques [11], and most recently, deep-learning [12].

Despite the successful progress of Deep-Learning in image classification, the high computational cost has been negatively affecting the development of video classification [13]. The performance of a visual recognition system is greatly influenced by the choice of visual features. Searching for the best feature representation is a priority to improve system performance in terms of accuracy. Based on this insight, researchers have presented several solutions to elicit the most important features through video streams. One approach, for instance, is the key frame-based method to find out the most relevant RGB frames in videos, which represent well the distinct actions for their corresponding videos and, consequently, decreasing learning scope [14].

In the domain of deep-learning neural networks, existing research methods are limited as they separate the representation of temporal information from the motion estimation techniques [15], which resulted in the need for more complex neural networks. Furthermore, existing methods depend on the assumption that the color of the moving points (pixel intensities) does not change among frames, which is not feasible in real situations. Due to such factors, there is a need to enhance the video features representation that is compatible with the principles of deep neural networks, searching for a common pattern between different examples within the same category.

This paper proposes a new feature-selection method that can capture relevant features only, while ignore irrelevant ones to recognize human activities in videos. The methodology assumes that each activity has its own movement-pattern that is well distinctive to be identified through activity recognition systems. To capture the basic pattern movements, we created what we intend to call a Growth method, which keeps track of the pattern's change through consecutive frames. This novel technique modeled motion features explicitly, while avoiding the negative effects of different body shapes, sizes, and other parts of irrelevant aspects that might affect the motion estimation.

The proposed methodology, in this paper, will simplify the neural network classifiers and drive it to process only relevant features to distinguish among different activities. Consequently, this new vision to capture motion patterns will minimize the complexity of designing deep-learning architectures to learn motion features.

The contribution of this research is to propose an efficient technique that extracts the basic movements for each activity through a Growth function that captures the shape of movement between two consecutive frames. The proposed technique is efficient since it achieves acceptable classification accuracy and minimizes the time and memory requirements.

This novel method of activity pattern representation is a suitable alternative to represent temporal information in videos instead of motion estimation techniques. The proposed video representation improves the learning process of deep neural networks of recognition systems in terms of accuracy, time, and memory consumption.

This paper is organized as follows: Section 2 highlights the contributions of this research among related work in the literature. Section 3 explains the proposed research methodology with detailed algorithmic descriptions. Section 4 presents our experiments to validate the proposed technique in terms of performance analysis, running time, and memory requirements. Finally, Section 5 concludes the research and discusses the future work.

2. Related Work

Video activity recognition is a time series classification task that requires combining motion features with video classification models into a machine learning system [16]. This

section tracks the progress in activity recognition research, motion representation, and architecture design to handle the learning of spatial and temporal information.

Donahue et al. [17] proposed LRCN (Long-term Recurrent Convolutional Networks). This model considers both RGB and optical flow images as inputs to their recognition system. The system follows the Two Stream networks methodology of Simonyan and Zisserman in [18] and utilizes the CNN-LSTM networks instead of the ConvNet classification model. Each video entered both networks at the same time in its two versions of RGB and optical flow. Consequently, the training of both networks learns the Spatial and Temporal features for each video and network. However, this approach is unable to capture wide-range temporal information.

Khan, S. et al. in [19] employed low-cost RGB-D sensors to combine skeletal data from RGB-D sensors with RGB and depth data. Using trained CNN on skeletal pictures as the fifth CNN stream resulted in high accuracy. Despite this, the approach supports the idea that as the amount of supporting evidence grows and recognition improves, this approach increases complexity to the recognition system in terms of extracting different views of information.

Ullah, A. et al. in [20] proposed two models that are able to classify full-length video streams. The first model is a Convolutional Temporal feature pooling that modifies a CNN for video recognition. The second model is a Long Short-Term Memory LSTM. The output of the underlying CNN is linked to the output of the LSTM layers at the class prediction score level. They additionally train both temporal models on optical flow images after training on RGB frames and perform late prediction fusion. The proposed method doubled the training time of both networks on two types of data besides using two separate networks, which considered time and memory consuming.

Tran et al. in [21] introduced an experimental design based on a Single Stream methodology. They applied 3D convolutions on a video volume using 3D filters. The research has applied different experiments to achieve the best 3D convolutional kernel and architecture that may achieve the best score of accuracy. They also discovered that the 3D convolutional networks (C3D) networks traced spatial information in the first few frames before the following movement in subsequent frames. However, the model was unable to capture long-range temporal features.

In order to accomplish a higher accuracy and learning convergence, Duta et al. in [22] introduced an improved version of the convolutional neural network (ResNets). Their technique allowed for the learning of exceptionally deep networks with over 400 layers (on ImageNet) and over 3000 layers (on CIFAR-10/100) with no optimization problems. The proposed building block contains four times more spatial channels in a building block than the original structure.

Feichtenhofer et al. in [23] developed a Convolutional Two-Stream network model that captures temporal characteristics by reducing long-range losses. Their contribution was to aggregate temporal neural networks' output across time frames to derive long-term dependency. They propose combining the networks at an early level so that the responses at the same pixel position are put in parallel rather than fusing at the end, like in the original two-stream structure. The researchers observed that spatially fusing both networks at the final convolutional layer was more accurate than fusing them earlier.

The concept of depth-based human action recognition, where the depth gives additional data to enhance the performance based on RGB frames, was studied by Gu et al. [24]. In this method, the recognition process is provided with depth information for additional motion patterns for human action. To extract the action pattern from the proposed depth-based motion history images (MHIs), the deep-learning model was used. This technique generated the MHI images with the depth information so the MHI could convert the motion history along with the depth directions instead of the single MHI from RGB images, since these can only characterize the motion bounded by the image plane. To capture the motion patterns, the CNN network of ResNet-101 was used. The study results showed that a better

performance can be achieved when the deep-learning model can discover discriminative characteristics from the depth MHIs of human actions.

The process of generating activity representation, based on pose estimation, is as suggested by Wang et al. in [25]. The pose is composed of several joints at the first layer, which are clustered and merged as spatial and temporal component sets, to model an action. This procedure is still a difficult problem due to its sensitivity to strong articulations, barely visible joints, occlusions, clothing, and lighting changes.

Other research in [26,27] provided efficient solutions for optimizing the classification process of human activities including aggregation of actions, pose recognition, and multi-modal sequences. There is a gap in maintaining running time and memory requirements.

3. Background Subtraction

Background subtraction is the process of separating the targeted moving objects from the static background. Since our work, in this research, is limited to videos from fixed-positioned cameras, the sequence of frames is expected to repeat the background on a large scale. Obviously, the background objects might be changed by time or have differed from one video to another. Therefore, in this section, we present our algorithmic technique to handle this issue. Furthermore, background model estimation depends on low-level pixel classification to produce an image with no moving objects, which must be kept regularly updated (not fixed). This is to adapt to the significant changes in the video frames related to geometry settings, illumination, weather, background motion, etc.

The Temporal Median Filter approach for background subtraction is adopted. This is to generate a dataset with a primary representation that will be used to produce the final feature representation of a clear activity pattern. The Temporal Median Filter approach is composed of four main processes: generating an initial background model; computing the difference of frames; applying binary thresholding; and updating the background model under the assigned learning rate. Table 1 provides a description for required parameters that will be used in the algorithmic description of the proposed methodology.

Table 1. Algorithmic Parameters.

Parameter	Explanation
stream	Stream of pixels that represents
Frames[ ]	Array of available frames
N	Number of frames in a given image
temp[ ]	Temporary array to hold a random number of frames.
(x,y)	Pixel coordination (2D)
t_i	Temporal time (round number i)
I	A frame that exists at median based on a given set of frames
D(x,y,t)	Distance between the background and the median at time t.
alpha	Threshold value

The initial background model (as shown in Algorithm 1) is formed by assuming that the pixel, which remains unchanged for over half of the time duration of the video, is considered as part of the background. The time threshold is empirical and based on several previous experiments.

Algorithm 1 Background Modeling Abstraction Algorithm

$bkg\ (stream)$:
1 $Input:\ Frames\ [N]$
2 *begin*
3 $temp[\,] = random(Frames)$
4 *For each frame* f *in* $temp\ [\,]$
5 *For each pixel*(x,y) *in* f
6 $bkg(x,y) = median\{I(x,y,t_i)\}, \forall (I \in\ N)$
7
8 $Output:\ I(x,y,t_i)$
9
10 *end*

Algorithm 1 depicts the procedural method to initiate the background by selecting N random frames and compute the median. The median (I), on the other hand, is a frame by itself. This process is illustrated in Figure 1 to show how the initial frame is selected.

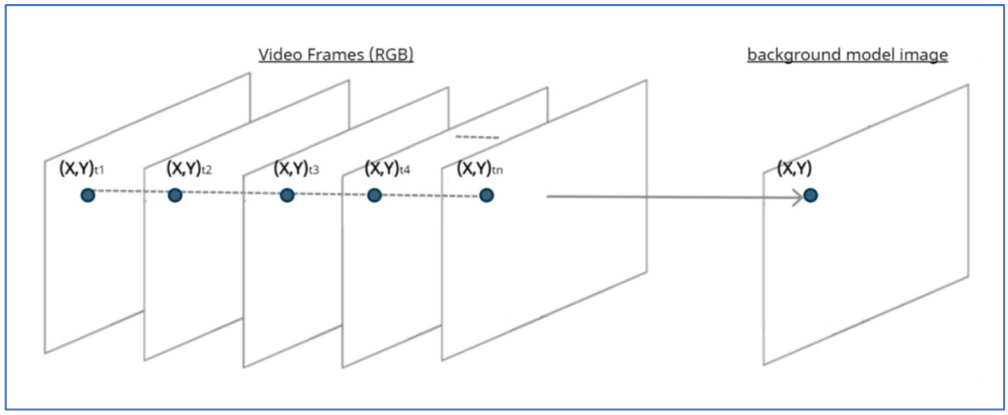

Figure 1. Simulating background modeling starting from a set of frames to induce the static background.

At this point, we can apply the temporal median approach in order to identify the background and then, remove it. Algorithm 2 illustrates the procedure to locate the background part using the initial model. The algorithm scans all frames after converting them into their gray-scale representation. The distance between the initial model and the given frame background is computed. Then, the resulting distance is compared to a threshold value in order to develop a binary image for each frame.

Algorithm 2 Temporal Median Filtering Algorithm

Temporal Median Filtering
1 $Input:\ Frames\ [N]$
2 *begin*
3 $t = 1$
4 $bkg\ (t)$
5 *while* $t < N$ *do*
6 $Grayscale\ (t)$
7 $D(x,y,t) = |\ I(x,y,t) - bcg(x,y,t)|$
8 $F(x,y,t) = \begin{cases} 1 & if\ D(x,y,t) > thresh \\ 0 & otherwise \end{cases}$
9 $\overline{bkg}(x,y,t) = alpha \times I(x,y,t) + (1 - alpha) \times \overline{bkg}(x,y,t-1)$
10 $t = t + 1$
11 *loop*
12 *end*

As stated in line eight, the array $F(x, y, t)$ is created for each frame to reflect a new binary representation. The binary image is produced by setting a threshold value for the pixel intensities of the gray-scale image to classify each pixel value into a foreground (white: 1) or background (black: 0) according to the threshold value.

Finally, binary images are passed to the next task in order to select the moving object's features, while ignoring the original frames.

4. Movement Feature Selection

The proposed technique aims to generate new images to extract the essence of the movement through consecutive frames to form an activity pattern. Therefore, each image encapsulates the moving parts of a given object. A major contribution of this research is the development of the function Growth, which tracks the development of movement, performs pixel-wise processing and run on each two consecutive binary frames resulting from the previous phase. In other words, the Growth function plays the role of a trajectory function for the corresponding physical activity. As a result, any human activity that contains movement of single or multiple body parts will be identified and extracted.

The Growth function aims to capture the development of movement through frames, according to the truth table presented in Table 2. The Growth function assumes that the activity pattern image is initially black. The black image is modified at a pixel location to white once the status of the activity at that location is moving, which means only at locations where the movement is currently passing by. The body part in its static status is not observed in the output image.

Table 2. The Truth Table of the Growth Function.

Current Frame Pixel Value at (x_t,y_t)	Next Frame Pixel Value at (x_{t+1},y_{t+1})	Activity Status Pixel Location (x,y)	Activity Pattern Location (x,y)
black (0)	black (0)	Not exist	black (0)
black (0)	white (1 or 255)	Moving	white (1 or 255)
white (1 or 255)	black (0)	Leaving	black (0)
white (1 or 255)	white (1 or 255)	static	black (0)

A simple illustration of how the growth function work on the binary images that has produced in the previous phase is explained as follows:

1. Pixel (X_t,Y_t) is black and pixel (X_{t+1},Y_{t+1}) is black: means the movement does not exist at this location. The corresponding location at the pattern image is still black.
2. Pixel (X_t,Y_t) is black and pixel (X_{t+1},Y_{t+1}) is white: means the activity is now moving in this location. The corresponding location in the pattern image is white.
3. Pixel (X_t,Y_t) is white and pixel (X_{t+1},Y_{t+1}) is black: means the movement is now leaving this location. The corresponding location in the pattern image is black.
4. Pixel (X_t,Y_t) is white and pixel (X_{t+1},Y_{t+1}) is white: means the movement is now static at this location. The corresponding location in the pattern image is black.

Figure 2 shows an example of applying the growth operation between two consecutive frames. The Growth operation extracts the parts that are currently moving. Frame_t represents an object at a time: (t) in blue. Frame_{t+1} represents the object that moved at the time: $(t+1)$ in blue. The output image represents the movement that occurred. The three locations in red in frame_t had no movement. The object moved in frame_{t+1} to three locations. Applying the growth operation between frame_t and frame_{t+1} have detected these three locations that had moved and recorded them on the black image. At every two consecutive frames, the growth function records the shape of movement at the time (t). Applying this method on every consecutive frame will allow for the development of a pattern of movements; activity.

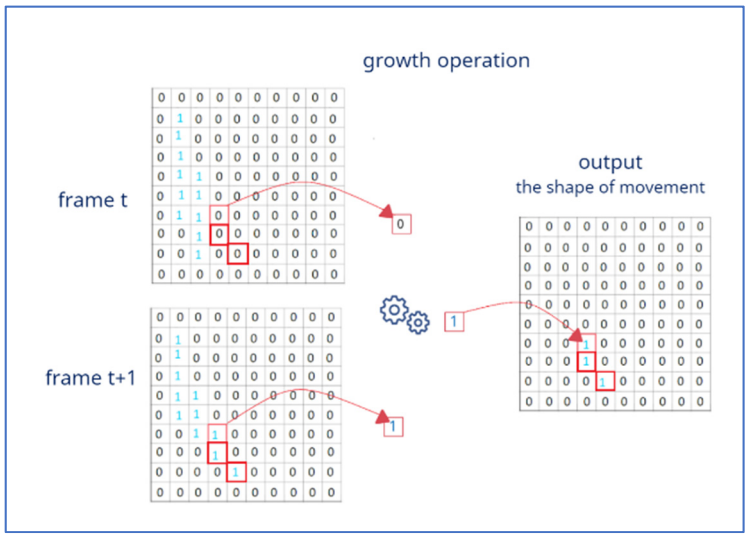

Figure 2. Semantic Explanation of the Growth Operation.

The growth operation captures the modifications of the activity pattern alone. If a part of the body had a role in performing the activity at a time and (t) had no role at a time $(t+1)$, then it disappears in the pattern images. Figure 3 show the behavior of the growth operation on the binary images of the swing bench activity.

Figure 3. The Application of the Growth Function on Swing Activity.

Figures 4 and 5 illustrate the movement pattern of the Jumping Jack sport as generated by the growth function. It shows how the function behaves to induce the actions that represent a pattern for the whole activity.

Figure 4. Original Activity Action frames.

The images of the movement pattern Figures 4 and 5 show the body parts that were involved in performing the activity. Moreover, the body parts appear relatively different in volume through the frames. This indicates distinct movements in performing the activity, in terms of speed and distance moved. Hands, legs, and the torso performed different movements to accomplish the activity. Therefore, they appear relatively different in the activity pattern images, which indicates the different ways these body parts were involved in the activity.

Figure 5. Action Frames after Applying the Growth Operation.

What distinguishes each activity in this representation is the shape of the movement itself, rather than the body shape or the displacement alone during the video stream. Furthermore, this research assumes that this shape of movement is important to learn during training recognition models. Since it contains the spatiotemporal information of the pixels related to the activity only.

Compared with the existing techniques of motion estimation such as Motion History, Optical Flow, and Poses Estimation, the proposed method produces the activity class by its nature. This can be an alternative to the RGB frames as an input to the activity recognition systems. In addition, the growth operation does not rely on the assumption that pixel intensities of an object remain constant between consecutive frames as optical flow images. The Growth function extracts how pixels behave in response to the activity rather than what the actual intensity is.

The following pseudo-code (Algorithm 3 and Algorithm 4) shows the algorithmic details of activity pattern generation using the Growth function:

Algorithm 3 Growth Function Pseudo-code

```
Function Growth returns the shape of movement between two Frames
    Growth_fun (current_frame, next_frame)
1       movement_shape_image < − − − − black
2       For i = 1 to m
3           For j = 1 to n
4               IF (current_frame [i, j] == Black   and   next_frame [i, j] ==  white)
5                   movement_shape_image [i, j] = white
6               End IF
7           End FOR
8       End FOR
9       Return movement_shape_image
10  End Function
```

Let m be the number of rows in the 2D matrix movement_(shape_image) that store a given image (frame) and n be the number of columns. Line 1 requires $O(1)$ to execute. Lines (4) and (5) require $O(2)$ to execute since both the condition and the assignment statements

require $O(1)$ for each. The inner loop (Lines 3 to 7) requires $O(n) \times O(2) \approx O(n)$. The outer loop (Lines 2 to 8) requires $O(m \times n)$. Finally, the return statement in Line (9) requires $O(1)$.

Therefore, the time complexity for the function Growth is defined as the sum of the time complexity of line (1), the outer-loop, and line (9). This is equal to:

$$O(1) + O(m \times n) + O(1) \approx O(m \times n)$$

Algorithm 4 Pattern Generation Algorithm

Generating Activity Pattern Images for a video of frame size n × m

```
    main ( )
1       Activity_Pattern_Images = [ ]
2       current_frame [n, m]
3       next_frame [n, m]
4       Shape_of_movement _image [n, m]
5       t = 1
6       While (current_frame! = last_frame)
7           current_frame < − − − − Frame(t)
8           next_frame < − − − − Frame(t + 1)
9           Shape_of_movement _Image = Growth_fun(current frame, next_frame)
10          Activity_Pattern_Images.append(Shape_of_movement _Image)
11          t = +2
12      End While
    End Program
```

Let the total number of frames in a 2D image is k. Lines 1, 2, 3, 4, and 5 require $O(5) \approx O(1)$ to execute. The while loop requires $O(k)$, where k is the total number of frames in a given video stream. Lines 7 to 11 require $O(1 + 1 + (m \times n) + log_{m \times n} + 1) \approx O(m \times n)$. Notice that $m \times n \gg \log_2$ for all $m \times n > 0$. This is due to Line 9 requiring the exact time complexity of Algorithm 3. Furthermore, Line 10 scans $\log_{\mathrm{m \times n}}$ every time. This implies that Algorithm 2 requires:

$$O(5) + [O(k) \times O(m \times n)] \approx O(k \times m \times n)$$

5. Experiments and Results

5.1. Datasets

The computer vision research center at the University of Central Florida (UCF) has developed a video-based action recognition dataset (UCF-101) that consists of 13,320 short videos. This collection depicts 101 different activities. The activities in UCF-101 belong to a large set of categories; making UCF-101 the most diverse dataset in this domain as compared to other benchmarks. The activities in the entire dataset can be classified into several groups: Human-Object Interaction, Only Body Motion, Human-Human interaction, Playing Musical Instruments, and Sports Videos. In this research, we chose the Sports category in order to generate activity patterns for each of its classes (8 classes) as shown in Table 3.

Table 3. Selected classes with number of videos per class.

	Classes	Number of Videos
1	Body Weight Squats	87
2	Boxing Punching Bag	114
3	Hula Hoop	69
4	Jumping jacks	93
5	Pommel Horse	49
6	Tennis Swing	101
7	Wall Pushups	102
8	Table Tennis Shot	111

Each activity class contains 25 s-long videos, and each video has been divided into 4–7 shorter ones. The videos for each class were kept with the same orientation, pose, viewpoint, and the same background as the original long video. Moreover, these videos have different time durations; thus, for training purposes, a fixed sequence length for the recognition network is required.

To glean a robust overview about the results of this research, we included two publicly available datasets; the KTH [28] and WVU [29] datasets. The KTH dataset contains six categories where each class is executed several times by 25 subjects during different scenarios. Each video in the KTH dataset is 160 × 120 spatial resolution and is 100 frames long. The WVU dataset has twelve different actions. Each sequence in this is for a subject who performs only one action. The spatial resolution of each video is 640 × 480 and is 71 frames long. For comparing the results with other methods only, we omit two classes included in the WVU datasets: namely, waving one hand and bowling.

5.2. Experiment Setup

Data preprocessing and experiments are implemented in Python 3.8.3 on Google Colab Pro platform using Graphics Processing Unit (GPU) at runtime. The experiments have been conducted on Google Colab Pro with GPU RAM limited to (25.46) GB, and time sessions limited to 24 h. Colab Pro virtual Machines (VMs) disk space is limited to (147.15) GB as well. It allows for accessing the fastest Colab GPUs such as to NVIDIA P100 or T4. The GPU is operating at a frequency of (1290) MHz, whereas memory is running at (876) MHz.

5.3. Experiment Results and Discussion

Two types of experiments have been conducted to analyze the effect of our proposed technique in terms of performance accuracies and computer resources (CPU and Memory) as compared to the baseline method; RBG. We applied both RBG and Growth methods to extract and select representative features for the purpose of recognizing activities in UCF-101, KTH and WVU datasets. Accordingly, we measured the performance of state-of-the-art classifiers using RBG and Growth methods. Furthermore, we measure the effect of applying the Growth method on the computer resources: CPU time and Memory space.

We have set the callback function to Early-Stopping, with a patience value equal to seven—this is to stop training if there were no improvements after the assigned number of epochs. Moreover, early stopping also prevented the model from over-fitting. The loss function that applied to calculate the prediction error was the categorical cross-entropy loss for multi-class classification. The optimization algorithm used in the network, to update the attributes' values to reduce the losses, was the Stochastic Gradient Descent (SGD). The network iterates through the batch and produces the prediction results. The difference between the actual value and the predicted value was computed as a loss, and then the optimizer continued adjusting the weights directed by the actual result.

5.3.1. Performance Indicators

During the training of the first experiment, videos were resized to 64 × 64 dimensions in all datasets. We trained the neural network with RGB videos of (71) sequence length to

align all datasets. Multiple k-filters were tested during several experiments. The network of 32 filters, two dropout layers with the rate of (0.5) and (0.5), respectively, to avoid over-fitting, a learning rate of (0.001), and dataset splitting (80% training, 20% testing) were adjusted for the RGB baseline experiment. During the training of the second experiment, videos were resized to 64 × 64 dimensions as well. We trained the neural network on videos of pattern images of (71) one-channel frames to align all datasets. Several numbers of filters were tested during several experiments. The network of 16 filters, two dropout layers with the rate of (0.2) and (0.3), respectively, to avoid over-fitting, a learning rate of (0.001), and dataset splitting (70% training, 30% testing) were adjusted for the Growth images experiment. Tables 4–6 depict the performance measurements (Precision, Recall, and F1-score) for both experiments using UCL, KTH and WVU datasets, respectively.

Table 4. RGB base (R) and Growth (G) Performance Analysis on UCL dataset.

	Precision (%)		**Recall (%)**		**F1-Score (%)**	
	R	**G**	**R**	**G**	**R**	**G**
Body Weight Squats	93	96	91	95	91	97
Boxing Punching Bag	85	87	99	99	90	94
Hula Hoop	99	100	81	90	88	93
Jumping jacks	58	96	100	100	75	97
Pommel Horse	99	95	82	94	90	94
Tennis Swing	99	96	100	100	100	98
Wall Pushups	98	99	73	100	82	100
Table Tennis Shot	100	100	90	94	94	97
Average Accuracy					89	96
Macro Average	92	96	89	97	88	96
Weighted Average	91	96	90	97	89	97

Table 5. RGB base (R) and Growth (G) Performance Analysis on KTH dataset.

	Precision (%)		**Recall (%)**		**F1-Score (%)**	
	R	**G**	**R**	**G**	**R**	**G**
Walking	93	97	87	97	89	96
Jogging	87	91	88	93	91	97
Running	94	99	86	98	91	97
Boxing	82	99	92	100	82	99
Handwaving	96	99	89	97	92	96
Handclapping	92	95	94	99	97	96
Average Accuracy	91	97	89	97	90	97

Table 6. RGB base (R) and Growth (G) Performance Analysis on WVU dataset.

	Precision (%)		**Recall (%)**		**F1-Score (%)**	
	R	**G**	**R**	**G**	**R**	**G**
Standing Still	90	91	88	94	89	92
Nodding head	89	96	88	98	89	97
Clapping	94	97	91	96	90	96
Waving 2 hands	87	98	91	100	89	99
Punching	91	98	90	96	91	98
Jogging	92	95	91	95	90	95
Jumping Jack	90	94	90	95	92	95
Kicking	90	96	86	97	89	97
Picking	85	93	89	95	87	96
Throwing	87	96	86	97	87	97
Average Accuracy	90	95	89	96	89	96

As shown in Tables 4–6, the performance indicators of the proposed feature-selection technique outperform the RGB baseline. Furthermore, we notice that the enhancement on recall measure was high. This indicates that the proposed growth-based pattern technique was able to recognize activities precisely as compared to the well-known RGB feature-selection technique. Moreover, we performed statistical analysis to study the resulting differences between both experiments. The T-test analysis (one-tailed with 0.05 significant level) is applied to infer whether the significance of the difference between the means of both samples. The results are as follow: $Mean = 0.0695$, $\frac{SS}{df} = 0.01$, $SM = 0.0289$, $and\ t = 2.25$. The value of p is 0.0294. Therefore, the results indicated that the difference is significant at $p < 0.05$. This implies that the proposed method achieved a significant enhancement with confidence of 95%.

Moreover, we ran the pools of features, which have been selected by both RGB-base and Growth-base techniques, on different state-of-the-art classification techniques that have been discussed in the literature section. Table 7 illustrates the resulted average accuracies.

Table 7. Comparison with State-of-the-Art methods using the Average Accuracy Measure.

Model	RGB Based (%)	Growth Patterns (%)	Improvement (%)
Conv-Lstm [17]	68.20	82.34	20.73
KcWKNN (on KTH) [19]	91.1	98.3	−1.0
KcWKNN (on WVU) [19]	92.4	98.7	−1.1
3D Convolutional [21]	82.30	90.40	9.84
Convolutional Two Stream Fusion [23]	92.50	94.20	1.84
Improved two streams architecture [25]	94.00	94.20	0.21
Two Stream Fusion Convolutional [26]	92.70	93.60	0.97
Convolutional (ResNet-101) [24]	67.96	84.44	24.25
Two Stream Fusion MLP- LSTM [30]	79.21	96.92	22.36
Convnet conv-Lstm [27]	75.40	77.90	3.32

As shown in Table 7, our proposed technique achieved an acceptable level of performance in terms of average accuracy in almost every state-of-the-art classifier in the domain of video-based activity recognition. Indeed, the growth method achieved such performance with a significantly smaller number of required features to identify activities. When compared to KcWKNN on KTH and WVU datasets, the Growth patterns achieved negative improvement. This could be due to the fact that the KcWKNN approach uses the fusion of two well-known deep learning methods (DenseNet201 and InceptionV3). The approach applies a feature selection based on kurtosis by using the fourth momentum. As a result, their method provided very high accuracies on KTH and WVU datasets.

Figure 6 depicts a trending analysis that explains the growth of the number of features as more datasets are used. The analysis shows a linear increase as compared to an exponential one for the RGB method. Furthermore, the R^2 value on each line in Figure 6 indicates the correlation of the trending-line to the original real data. In the case of the growth-based technique, $R^2 = 0.79$ is a strong positive relationship, which indicates that the trending line shows a realistic correlation with the original data.

To conclude, the performance indicators showed that our proposed feature-selection method achieved a significant goal: achieving an acceptable accuracy as compared to the state-of-the-art techniques, while requiring less feature dimensionality. In fact, the utmost goal of our proposed method is maintaining high accuracy, while minimizing the pressure on computer resources: CPU and RAM. Moreover, the proposed growth-pattern method is compatible with the deep learning principles of learning common patterns of a given class. Therefore, the quality of features in the pattern images is more powerful for the learning process, as compared to the features in original images that contain the whole scene features relevant to the activity and the irrelevant ones.

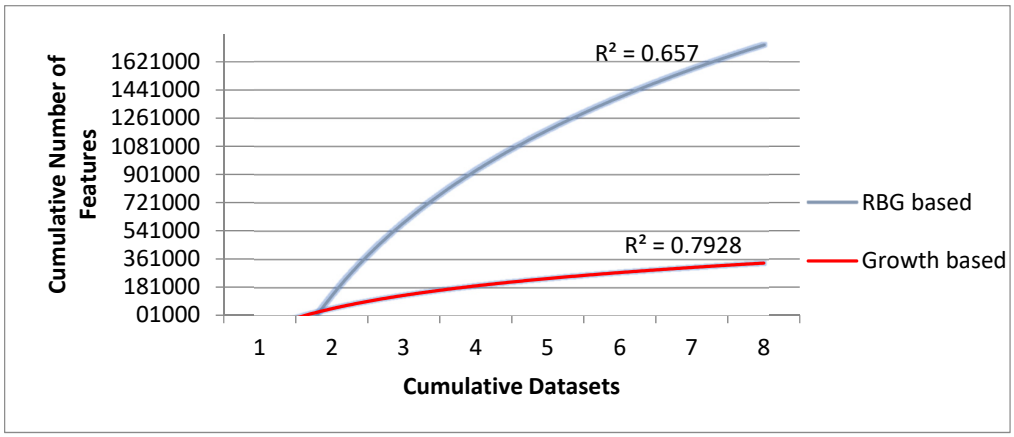

Figure 6. Trending Analysis of how the number of features grow as the number of datasets increase.

5.3.2. Running Time and Memory Analysis

We performed additional experiments to measure the effect of the proposed method on the classification task in terms of actual running time and memory requirements. The experiments have been conducted on Google Colab Pro with GPU RAM limits to (25.46) GB, and time sessions limit to 24 h. Colab Pro VMs also generally come with double the disk of standard Colab VMs limits to (147.15) GB Disk space. Colab Pro allows to access the fastest Colab GPUs such as to NVIDIA P100 or T4. The GPU is operating at a frequency of (1290) MHz (the speed of the GPU's chip), whereas memory is running at (876) MHz (the speed of the VRAM on the GPU).

We conducted these experiments to show such effects as the number of different input videos increased gradually. First, we ran the classification model on small dataset of 50 videos and then, increased the number by 50 videos each time. Consequently, this would show us how our proposed method performs as the size of input data increases gradually. Table 8 shows the experimental results.

Table 8. Incremental Analysis for Computer Resources Utilization.

Number of Videos	Collection Size	RGB		Growth Patterns	
		Running Time (Seconds)	Memory Allocation (GB)	Running Time (Seconds)	Memory Allocation (GB)
50	321 MB	4.958	5.41	50	321 MB
100	670 MB	5.21	5.5	100	670 MB
150	973 MB	7.51	5.75	150	973 MB
200	1.23 GB	9.67	7.07	200	1.23 GB
250	1.58 GH	12.1	8.57	250	1.58 GH
300	1.98 GB	14.4	10.07	300	1.98 GB
350	1.89 GB	16.6	8.58	350	1.89 GB
400	2.51 GB	18.8	8.59	400	2.51 GB

Figure 7 shows the actual GPU running time. During this experiment, we noticed that at small datasets, the actual running of the proposed method is approaching one second constantly (at 50, 100, and 150 videos). On the other hand, at small datasets, the actual GPU time increases by a polynomial fashion. Another interesting outcome is that: as the number of videos increases, the GPU time increases in a sub-linear time when applying the proposed method. While it increases in a polynomial shape during the running of the original RBG. Such data leads us to conclude that the reduction in the number of features, which has resulted from applying the proposed method, has a positive effect in terms of actual running time.

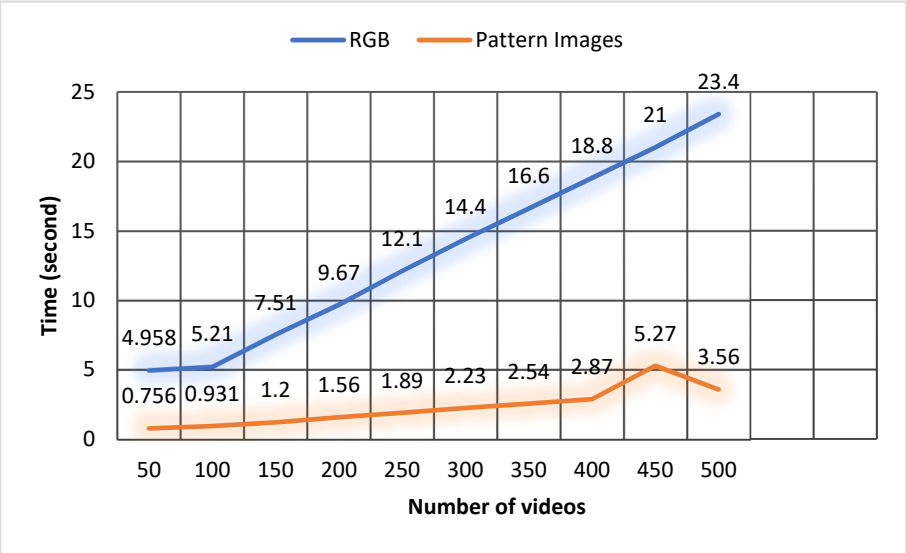

Figure 7. Actual Running Time Analysis as the number of videos increase.

Figure 8 shows how the memory (RAM) is affected by applying the proposed method. According to the results, the proposed method utilizes an acceptable and nearly constant amount of memory; especially at large datasets (above 200). On the other hand, there was a linear increase in the size of required memory during the running of RBG data features. This clearly shows how the feature-reduction method minimizes the amount of required memory.

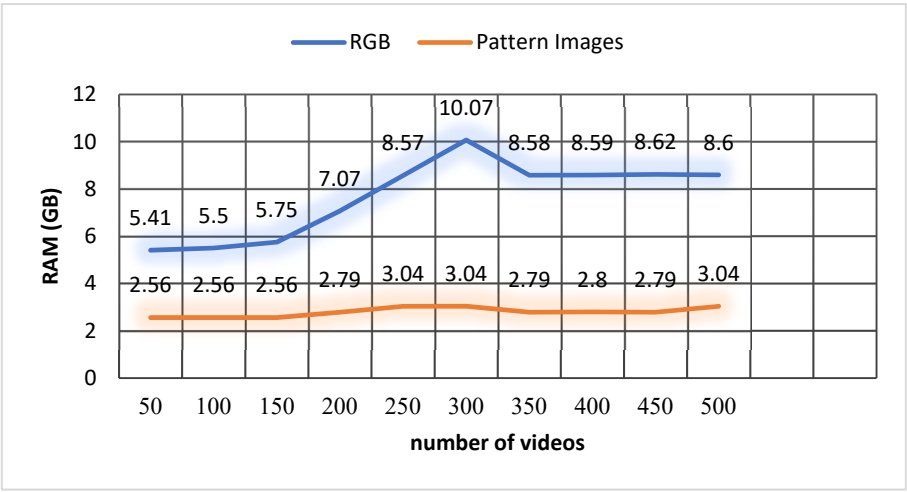

Figure 8. Actual Memory (RAM) Utilization as the number of videos increase.

6. Conclusions and Future Work

This paper presented a novel feature-selection approach for activity recognition. The proposed technique (Growth Function) extracts the shape of movement from two consecutive frames; generating a new video representation containing the spatiotemporal information needed for a classification task called activity pattern images. The pattern images are representative enough to be applied to vision tasks involving activity recognition, similarity analysis of video sequences, and other video applications. Moreover, the pattern images hold the least amount of data that distinguish each activity from other ones. Consequently, experiments showed promising results for video activity recognition of a stable camera.

We compared the results of two classification experiments in terms of recognition accuracy. The results proved that the proposed technique increased the accuracy of the activity recognition system compared to the baseline accuracy. In addition, the experimental results showed significant and promising enhancements as compared to other existing approaches that followed Single Stream and Two Stream networks for video activity recognition. Additional experiments were conducted to measure the efficiency of the research technique in terms of GPU running time and memory allocation, when performing classification for input data size that increased gradually. According to the results, the proposed method of feature reduction utilizes an acceptable amount of memory and has a positive effect in terms of classification time.

Author Contributions: Conceptualization, N.T., M.A.O. and A.K.B.; methodology, N.T., M.A.O. and M.G.A.Z.; software, N.T., A.K.B. and M.G.A.Z.; validation M.A.O., M.R. and M.G.A.Z.; formal analysis M.A.O., M.G.A.Z. and M.R.; investigation, N.T., M.A.O. and A.K.B.; writing-original draft preparation, N.T., M.A.O., M.R., A.K.B. and M.G.A.Z.; review and editing, M.A.O. and M.G.A.Z.; visualization, A.K.B., M.G.A.Z. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Avci, A.; Bosch, S.; Marin-Perianu, M.; Marin-Perianu, R.; Havinga, P. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In Proceedings of the 23th International Conference on Architecture of Computing Systems, Hannover, Germany, 22–23 February 2010; VDE: Berlin, Germany, 2010; pp. 1–10.
2. Al Zamil, M.G.; Samarah, S.; Rawashdeh, M.; Hossain, M.S.; Alhamid, M.F.; Guizani, M.; Alnusair, A. False-Alarm detection in the Fog-Based internet of connected vehicles. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7035–7044. [CrossRef]
3. Dragan, M.A.; Mocanu, I. Human activity recognition in smart environments. In Proceedings of the 2013 19th International Conference on Control Systems and Computer Science, Bucharest, Romania, 29–31 May 2013; IEEE: Manhattan, NY, USA, 2013; pp. 495–502.
4. Zamil, M.G.A.; Samarah, S.; Rawashdeh, M.; Karime, A.; Hossain, M.S. Multimedia-oriented action recognition in Smart City-based IoT using multilayer perceptron. *Multimed. Tools Appl.* **2019**, *78*, 30315–30329. [CrossRef]
5. Khurana, R.; Kushwaha, A.K.S. Deep Learning Approaches for Human Activity Recognition in Video Surveillance—A Survey. In Proceedings of the 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018; IEEE: Manhattan, NY, USA, 2018; pp. 542–544.
6. Alshboul, Y.; Bsoul, A.A.R.; Zamil, M.A.; Samarah, S. Cybersecurity of Smart Home Systems: Sensor Identity Protection. *J. Netw. Syst. Manag.* **2021**, *29*, 22. [CrossRef]
7. Gupta, S. Deep learning based human activity recognition (HAR) using wearable sensor data. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100046.
8. Huang, T.; Yang, G.; Tang, G. A fast twodimensional median filtering algorithm. *IEEE Trans. Acoust. Speech Signal Processing* **1979**, *27*, 13–18. [CrossRef]
9. Samarah, S.; Zamil, M.G.A.; Rawashdeh, M.; Hossain, M.S.; Muhammad, G.; Alamri, A. Transferring activity recognition models in FOG computing architecture. *J. Parallel Distrib. Comput.* **2018**, *122*, 122–130. [CrossRef]
10. Al Zamil, M.G.; Samarah, S. Application of design for verification to smart sensory systems. In *Qatar Foundation Annual Research Conference Proceedings Volume 2014 Issue 1 (Vol. 2014, No. 1, p. ITPP0366)*; Hamad bin Khalifa University Press (HBKU Press): Doha, Qatar, 2014.
11. Kumar, S.S.; John, M. Human activity recognition using optical flow based feature set. In Proceedings of the 2016 IEEE International Carnahan Conference on Security Technology (ICCST), Orlando, FL, USA, 24–27 October 2016; IEEE: Manhattan, NY, USA, 2016; pp. 1–5.
12. Nunez, J.C.; Cabido, R.; Pantrigo, J.J.; Montemayor, A.S.; Velez, J.F. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognit.* **2018**, *76*, 80–94. [CrossRef]
13. Justus, D.; Brennan, J.; Bonner, S.; McGough, A.S. Predicting the computational cost of deep learning models. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; IEEE: Manhattan, NY, USA, 2018; pp. 3873–3882.
14. Rachmadi, R.F.; Uchimura, K.; Koutaki, G. Video classification using compacted dataset based on selected keyframe. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), Singapore, 22–25 November 2016; IEEE: Manhattan, NY, USA, 2016; pp. 873–878.

15. Zamil, M.G.A. Multimodal daily activity recognition in smart homes. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23–26 April 2019; IEEE: Manhattan, NY, USA, 2019; pp. 922–927.
16. Nadi, R.A.; Zamil, M.G.A. A profile based data segmentation for in-home activity recognition. *Int. J. Sens. Netw.* **2019**, *29*, 28–37. [CrossRef]
17. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
18. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *arXiv* **2014**, arXiv:1406.2199.
19. Khan, S.; Khan, M.A.; Alhaisoni, M.; Tariq, U.; Yong, H.-S.; Armghan, A.; Alenezi, F. Human Action Recognition: A Paradigm of Best Deep Learning Features Selection and Serial Based Extended Fusion. *Sensors* **2021**, *21*, 7941. [CrossRef] [PubMed]
20. Ullah, A.; Muhammad, K.; Del Ser, J.; Baik, S.W.; de Albuquerque, V.H.C. Activity recognition using temporal optical flow convolutional features and multilayer LSTM. *IEEE Trans. Ind. Electron.* **2018**, *66*, 9692–9702. [CrossRef]
21. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
22. Duta, I.C.; Liu, L.; Zhu, F.; Shao, L. Improved residual networks for image and video recognition. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; IEEE: Manhattan, NY, USA, 2021; pp. 9415–9422.
23. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.
24. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Chen, T. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
25. Wang, C.; Wang, Y.; Yuille, A.L. An approach to pose-based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 915–922.
26. Girdhar, R.; Ramanan, D.; Gupta, A.; Sivic, J.; Russell, B. Actionvlad: Learning spatio-temporal aggregation for action classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 971–980.
27. Verma, P.; Sah, A.; Srivastava, R. Deep learning-based multi-modal approach using RGB and skeleton sequences for human activity recognition. *Multimed. Syst.* **2020**, *26*, 671–685. [CrossRef]
28. Laptev, I. Local Spatio-Temporal Image Features for Motion Interpretation. Ph.D. Thesis, Department of Numerical Analysis and Computer Science (NADA), KTH, S-100 44, Stockholm, Sweden, 2004.
29. Hassan, M.M.; Uddin, M.Z.; Mohamed, A.; Almogren, A. A robust human activity recognition system using smartphone sensors and deep learning. *Future Gener. Comput. Syst.* **2018**, *81*, 307–313. [CrossRef]
30. Raj, B.N.; Subramanian, A.; Ravichandran, K.; Venkateswaran, D.N. Exploring techniques to improve activity recognition using human pose skeletons. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, Colorado, AZ, USA, 2–5 March 2020; pp. 165–172.